PASSED

Computer lab 5, Group 4

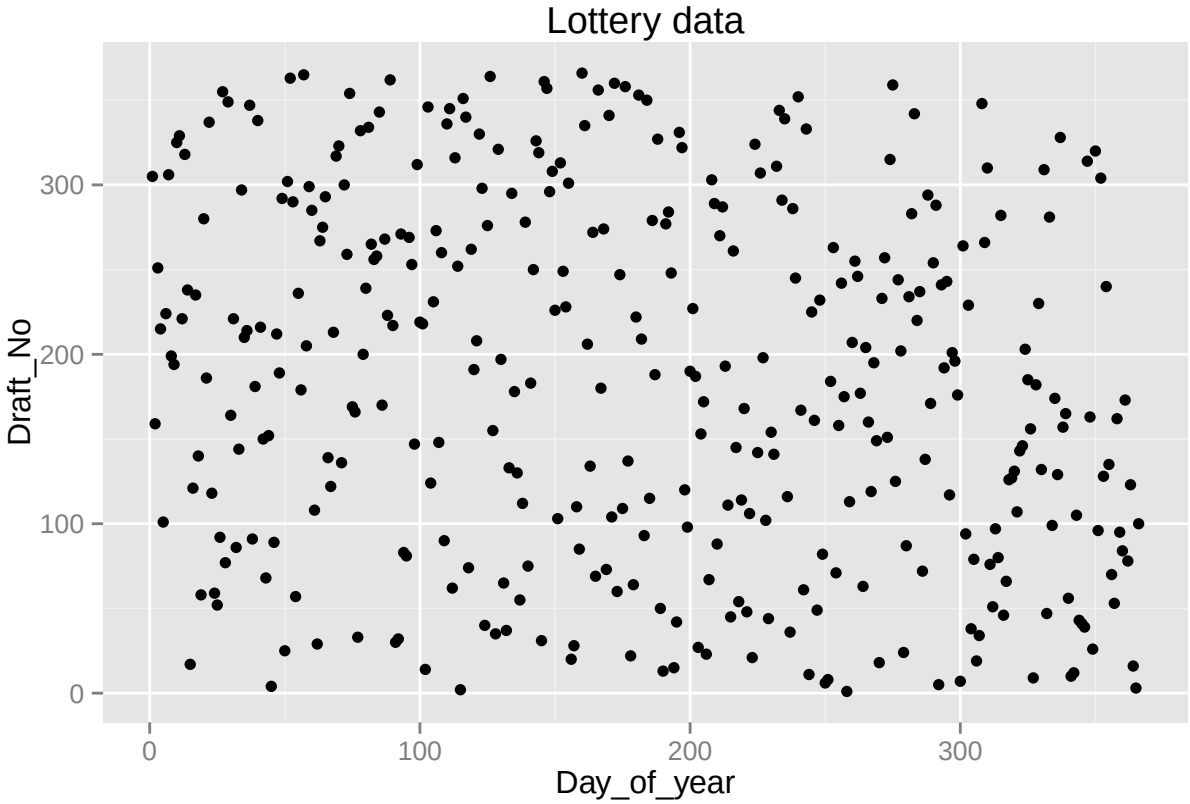
Kevin Neville, Gustav Sternelöv, Vuong Tran

10 mars 2016

Assignment 1: Hypothesis testing

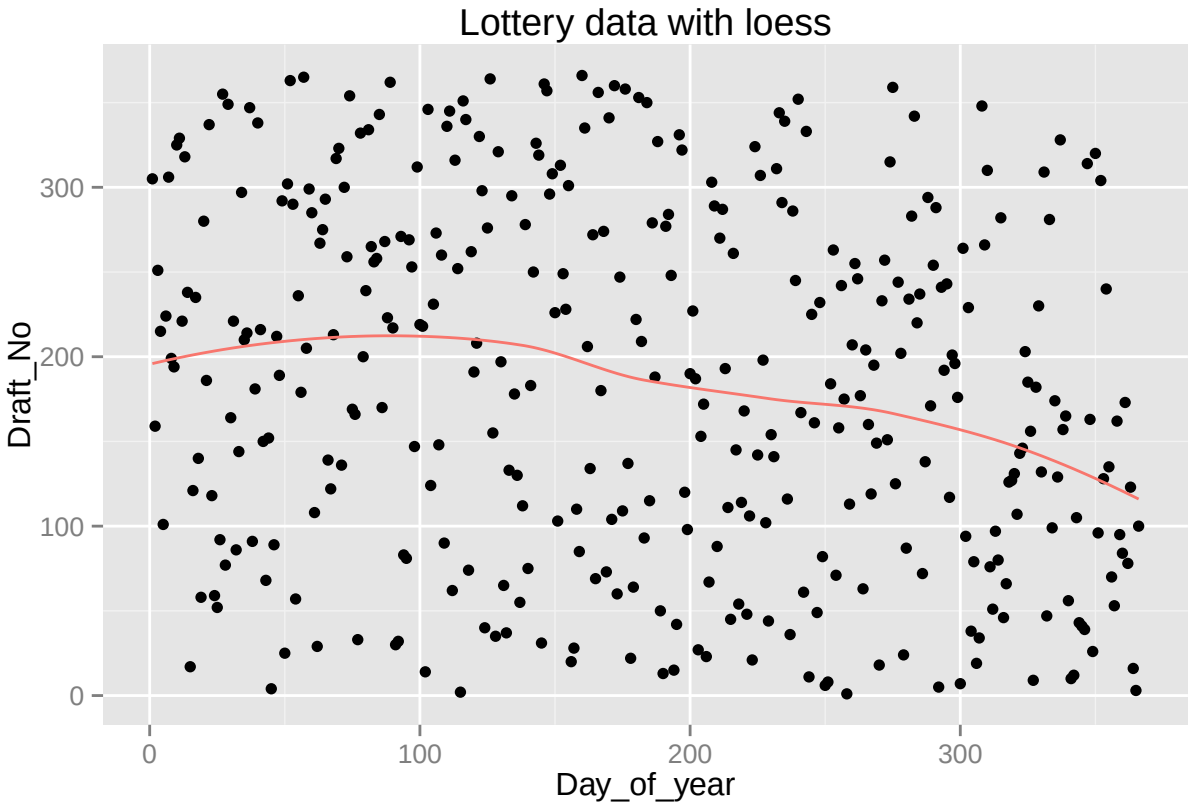
In 1970, the US Congress instituted a random selection process for the military draft. All 366 possible birth dates were placed in plastic capsules in a rotating drum and were selected one by one. The first date drawn from the drum received draft number one, the second date drawn received draft number two, etc. Then, eligible men were drafted in the order given by the draft number of their birth date. In a truly random lottery there should be no relationship between the date and the draft number. Your task is to investigate whether or not the draft numbers were randomly selected. The draft numbers (Y=Draft_No) sorted by day of year (X=Day_of_year) are given in the file lottery.xls

1.1. Make a scatterplot of Y versus X and conclude whether the lottery looks random.



By looking at the scatterplot it is hard to say anything else than that the lottery looks random. No trend or any other non-random pattern is thought to be seen in the plot. Instead, the values seem to be rather evenly spread out over the whole graph.

1.2. Compute an estimate $\hat{Y}$ of the expected response as a function of X by using a loess smoother (use loess()), put the curve $\hat{Y}$ versus X in the previous graph and state again whether the lottery looks random.



According to the loess model we seem to have declining trend. This would suggest that the lottery is not behaving randomly. At least it creates uncertainty for the lottery procedures that might be intresering to investigate further.

1.3. To check whether the lottery is random, it is reasonable to use test statistics.

$$T = \frac{\hat{Y}(X_b) - \hat{Y}(X_a)}{X_b - X_a},\ where\ X_b = arg\ max_x \hat{Y},\ X_a = arg\ min_x \hat{Y}$$

If this value is significantly greater than zero, then there should be a trend in the data and the lottery is not random. Estimate the distribution of T by using a non-parametric bootstrap with B = 2000 and comment whether the lottery is random or not. What is the P-value of the test?

We testing for randomness, we get the following hypothesis: When doing the hypothesis testing we use the most common significane level of 0.05.

$$H_0 : t \leq 0$$
$$H_a : t > 0$$

Since we obtain a very low p-value (0.0005) we can reject the null hypothesis of the lottery being random.

1.4 Implement a function depending on the data and B and that tests the hypothesis H_0: Lottery is random vs H_a: Lottery is not random by using a permutation test with statistics T and returns the p-value of this test. Test this function on our data with B=2000.

When using the non-parametric boostrap we obtain a p-value of 0.16. Since this is larger than 0.05 we can not reject the null hypothesis of the lottery being random.

1.5 Make a crude estimate of the power of the test constructed in step 4:

a. Generate (an obviously non-random) dataset with n= 366 observations by using same X as in the orginal dataset and Y(x) = max(0, min(α x + β, 366)) where α = 0.1 and $\beta \sim$ N(183, sd =10).

b. Plug these data into the permutation test with *B = 200* and note whether it was rejected.

c. Repeat steps a), b) for α= 0.2, 0.3 .. 10.

```
##   [1] 0.000 0.000 0.000 0.000 0.000 0.005 0.000 0.000 0.000 0.000 0.000
##  [12] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##  [23] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##  [34] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##  [45] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##  [56] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##  [67] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##  [78] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
##  [89] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## [100] 0.000
```

```
## [1] 1
```

The first value above is obtained when α=0.1. Since this p-value is very small we reject the null hypothesis. This is also true for all other p-values for the different α.

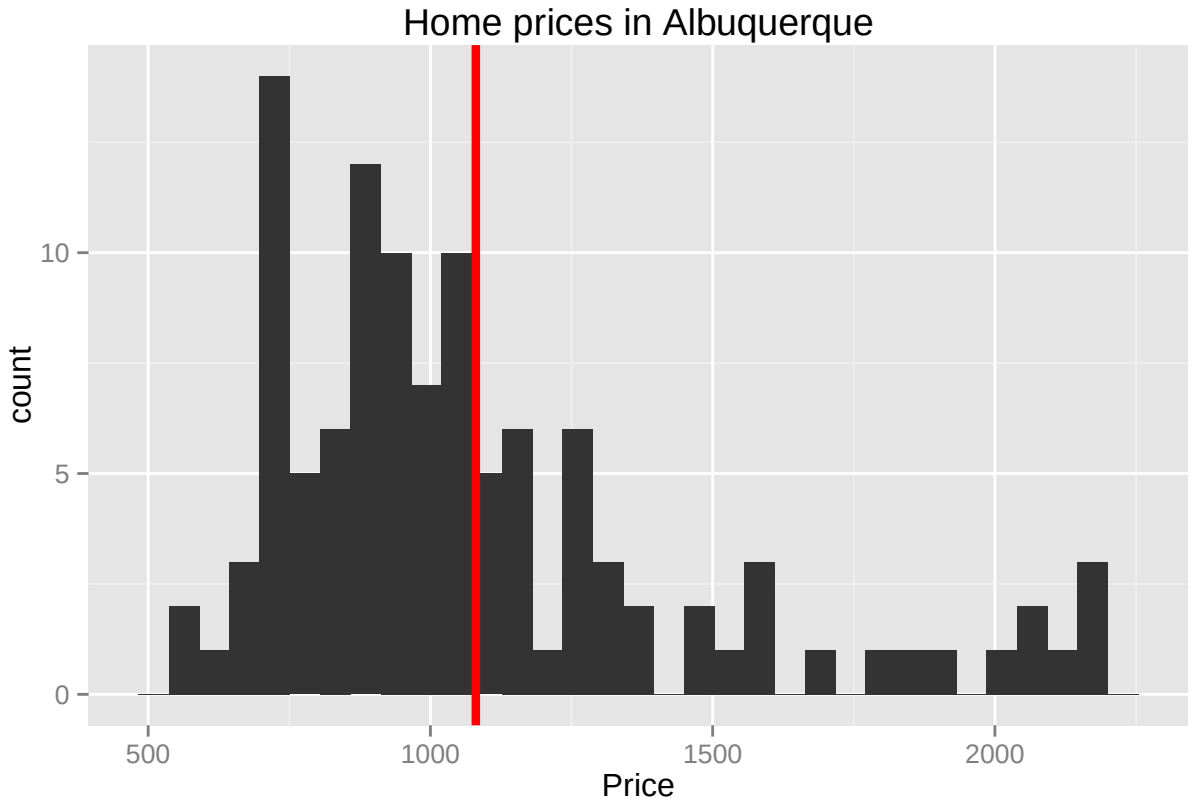
What can you say about the quality of your test statistics considering the value of the power?

Since the null hypothsies was rejected in all cases, the power equals one. Hence the quality of the test statistics is perfect.

Assignment 2: Bootstrap, jackknife and confidence intervals

The data you are going to continue analyzing is the database of home prices in Albuquerque, 1993. The variables present are Price; SqFt – the area of a house; FEATS – number of features such as dishwasher, refrigerator and so on; Taxes – annual taxes paid for the house. Explore the file prices1.xls

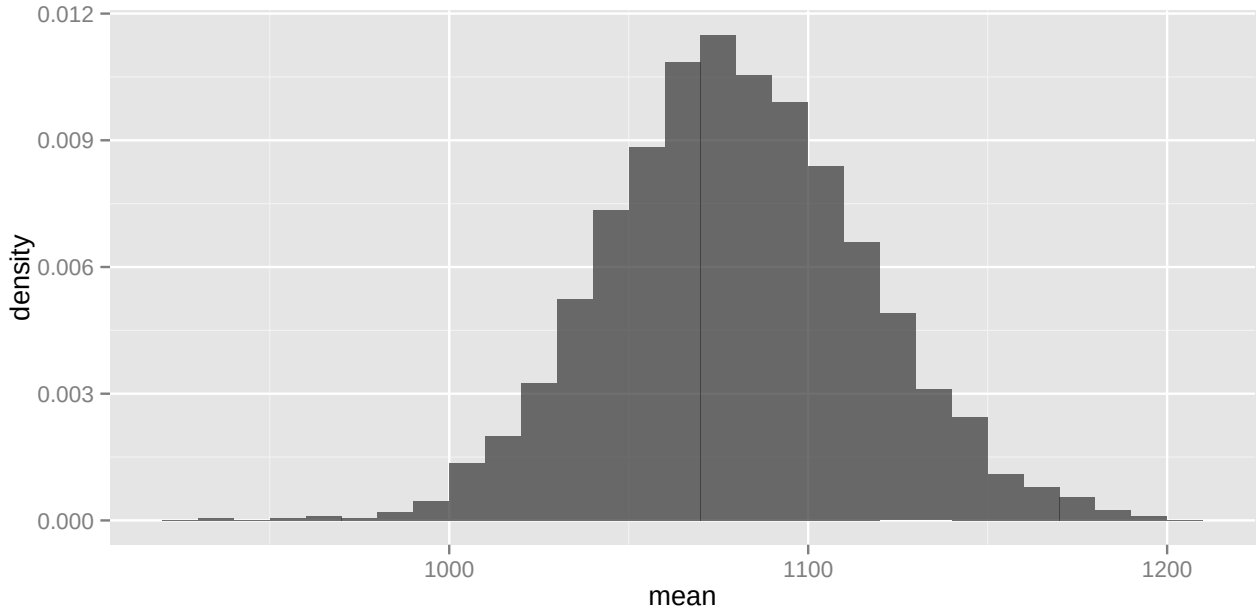
2.1. Plot the histogram of Price. Does it remind any conventional distribution? Compute the mean price.



The histogram does not remind us of any particular distribution, but of the most known distribution we would say it is closest to the chi-square distribution. The mean of the house prices is 1080.4727273.

2.2. Estimate the distribution of the mean price of the house using bootstrap. Determine the bootstrap bias-correction and the variance of the mean price. Compute the 95% confidence interval for the mean price using bootstrap percentile, bootstrap BCa, and first-order normal approximation (Hint: *use boot(),boot.ci(),plot.boot(),print.bootci()).*

A non-parametric boostrap is choosen since we are not certain about the true distribution.



The bootstrap bias-correction is obtained by first calculating the mean for the whole sample, 1080.47. Then, the mean of all the bootstrap means is computed, 1079.98. The bootstrap bias correction then is 2∗1080.47 -1079.98 = 1080.96.

The variance of the price of the mean is obtained by using a non-parametric bootstrap for the obtained estimates of mean and calculating the variance for each bootstrap sample. Again, B is set to 2000 and the estimated variance of the price of the mean is 1283.76.

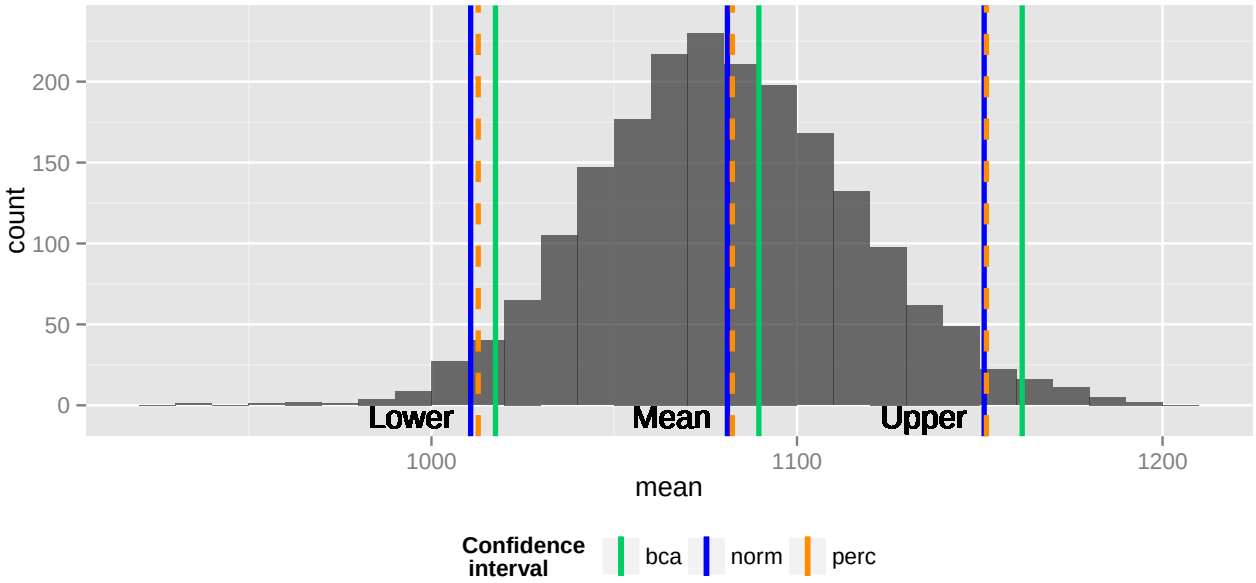
The table below shows the respective confidence intervals and lengths.

```
##      Lower    Upper   Length Method
## 1 1010.722 1151.204 140.4822   norm
## 2 1012.792 1151.817 139.0250   perc
## 3 1017.513 1161.607 144.0932    bca
```

2.3. Estimate the variance of the mean price using the jackknife and compare it with the bootstrap estimate.

The estimated variance of the mean when using the jackknife method is 1320.91. When comparing the obtained variance with the bootstrap estimate of the variance, we see that the jackknife variance is larger. This is expected since the jackknife method generally overestimates the variance.

2.4. Compare the confidence intervals obtained with respect to their length and the location of the estimated mean in these intervals.



The length of the estimated intervals are similiar for all three methods. However the *percentile* method is the smallest of the three, but only marginally. The *bca* method have the largest length of the three intervals. The normal method estimate of the mean is the closest one to the true mean. The bca method performs worst in both estimating the confidence interval with respect to its length, and also its estimate of the mean since this is the one which is most far off the true mean.

Contributions

All of the group members have contributed to this lab report. The code is a mixture of all the group members own code. The majority of the text is compiled together.

Appendix - R-code

```
library(gdata)
library(ggplot2)
lottery <- read.xls("/Users/Kevin/Desktop/Computational-statistics/Lab5/lottery.xls")
p1 <- ggplot(lottery, aes(y=Draft_No, x=Day_of_year)) + geom_point()
p1 + labs(title="Lottery data") # Yes looks random
# 1.2
loess_model <- loess(Draft_No ~ Day_of_year, data = lottery)
lottery$fitted_val <- loess_model$fitted
p1 + geom_line(data=lottery, aes(y=fitted_val, x = Day_of_year, color="red")) + labs(title="Lottery data
library(boot)
stat1<-function(data,n){
  data1=data[n,]
  loessM <- loess(Draft_No ~ Day_of_year, data=data1)
  Xa <- which.min(loessM$fitted)
  Xb <- which.max(loessM$fitted)
  y_Xa <- loessM$fitted[Xa]
  y_Xb <- loessM$fitted[Xb]
  T_val <- (y_Xb - y_Xa)/(data1$Day_of_year[Xb] - data1$Day_of_year[Xa])
  ret <- T_val
  return(ret)
}
set.seed(311015)
res1=boot(lottery,stat1,R=2000)
res1Dat <- data.frame(x=res1$t, index=1:2000)
options(scipen=999)
#1.4
#data should be a dataframe with X,Y column
MyPermTestFunc<-function(data,B){
  data$x<-data[,1]
  data$y<-data[,2]
  stat=numeric(B)
  n=length(data[,1])
  #t0 critical value
  estY0<-loess(y~x,data=data)
  xb0<-which.max(estY0$fitted)
  xa0<-which.min(estY0$fitted)
  t0<-(estY0$fitted[xb0]-estY0$fitted[xa0])/(data$x[xb0]-data$x[xa0])
  for(b in 1:B){
    Gb=sample(data$x,n)
    #our permutation set
    dataDf<-data.frame(X=Gb,Y=data$y)
    estY<-loess(Y~X,data=dataDf)
    xb<-which.max(estY$fitted)
    xa<-which.min(estY$fitted)
```

```
    yxb<-estY$fitted[xb]
    yxa<-estY$fitted[xa]
    #calculating Test statistica
    stat[b]=(yxb-yxa)/(dataDf$X[xb]-dataDf$X[xa])
  }
  #calculating p-value
  count<-0
  for(i in 1:length(stat)){
    if(abs(stat[i])>abs(t0)){
      count=count+1
    }
  }
  pvalue<-count/length(stat)
  return(pvalue)
}
set.seed(12345)
pvalue_14<-MyPermTestFunc(lottery[,4:5],2000)
#1.5
#a,b and c
alpha<-seq(0.1,10,0.1)
newdata<-lottery
PVec<-c()
#set.seed(12345)
for(i in 1:length(alpha)){
  for(j in 1:length(newdata$Draft_No)){
    beta<-rnorm(1,183,10)
    newdata$Draft_No[j]=max(0,min(alpha[i]*newdata$Day_of_year[j]+beta,366))
  }
  newdata[,4:5]
  PVec[i]<-MyPermTestFunc(newdata[,4:5],200)

}
PVec
powersum<-0
#reject ho? if p values<=0.05 then reject
for(i in 1:length(PVec)){
  if(PVec[i]<=0.05){
    powersum=powersum+1
  }
}
Power=powersum/length(PVec)
Power
price <- read.xls("/Users/Kevin/Desktop/Computational-statistics/Lab5/prices1.xls")

hist3 <- ggplot(data = price, aes(x=Price)) + geom_histogram() +
   geom_vline(aes(xintercept=mean(Price)), color="red", size=1.5) + labs(title="Home prices in Albuquer
hist3
# 2.2
## Function for estimating the mean value
stat3<-function(data,n){
  data1=data[n,]
  res = mean(data1$Price)
  return(res)
```

```
}
set.seed(12345)
res3=boot(price,stat3,R=2000)
priceBoot <- data.frame(mean = res3$t, index = 1:2000)
ggplot(priceBoot, aes(mean,..density..)) + geom_histogram(binwidth=10, alpha=0.7)
# Function for estimating the variance of the price of the mean
varBoot <-function(data,n){
  data1=data[n]
  res = (sum((data1 - mean(data1))^2)) * (1/(length(data1)-1))
  return(res)
}
set.seed(311015)
res4=boot(res3$t,varBoot,R=2000)
# 95 % C.I for the mean
CI <- boot.ci(res3, type=c("norm","perc", "bca"))
CIvals <- data.frame(rbind(CI$normal[2:3], CI$perc[4:5],CI$bca[4:5]), X3=c(CI$normal[3]-CI$normal[2], C
names(CIvals) <- c("Lower", "Upper", "Length", "Method")
CIvals
## 2.3
T_star <- 0
for(j in 1:110){
  T_star[j] <- 110*mean(price[,1]) - 109 * mean(price[-j, 1])
}
J_T <- (1/110) * sum(T_star)
varJack <- sum((T_star-J_T)^2) / (110*109)
CIvals$Mean <- (CIvals$Lower+CIvals$Upper)/2

ggplot(priceBoot, aes(mean)) + geom_histogram(binwidth=10, alpha=0.7) + geom_vline(data=CIvals, aes(xin
  scale_linetype_manual(name="", values=c("solid", "solid","dashed"), guide=FALSE) +
  theme(legend.position = "bottom") +
  geom_text(aes(CIvals$Lower[1],0,label = "Lower", vjust = 1, hjust=1.2))+
  geom_text(aes(CIvals$Upper[1],0,label = "Upper", vjust = 1, hjust=1.2))+
  geom_text(aes(CIvals$Mean[1],0,label = "Mean", vjust = 1, hjust=1.2))+ ylim(-7, 235)
## NA
```